A Closed Form Solution to the Reconstruction and Multi-View Constraints of the Degree d Apparent Contour

David McKinnon[1], Barry Jones[2] and Brian C. Lovell[1]
[1]School of Information Techology and Electrical Engineering
Intelligent Real-Time Imaging and Sensing (IRIS) Group
University of Queensland
[2]Department of Mathematics
University of Queensland

Abstract

This paper presents a novel theoretical approach to calculating the apparent contour of a smooth surface. The problem is formulated as a dual space intersection of algebraic tangent cones, which we will consider to be the members of degree d hypersurfaces. The well known thereotical foundation for multi-view geometry is extended in light of this to solve the problems of triangulation and forming multi-view matching constraints for degree d apparent contours.

1 Introduction

The problem of reconstructing a static scene from mutliple images is rich field of research [3]. There are a wide range of algorithms that can be used to reconstruct linear features such as points and lines from there projections in mutliple images as well as the inverse problem of finding the egomotion of the camera observing the scene.

However there has been far less research on the reconstruction and multiple view geometry of arbitrary curves observed in the scene [5], although the simplest cases of the conic and quadric have been investigated to a greater extent [4, 6, 1].

Broadly speaking, there are two different catagories of curves commonly observed in a scene, these are the *static* and *apparent contours*. Static curves are rigid curves they may commonly occur as textures on a surface or a thin thread of wire or other such objects. Each point on a static curve obeys the regular epipolar transfer equations, however presently there is only one approach to finding a closed form algebraic solution for their geomerty in the general degree d case [5] and several for the special degree 2 case [4, 6, 1].

We say the degree 2 case of the static curve is a special case since it is the only type of 3D algebraic curve that can be described by one equation. The other class of geometric objects that can be described by one equation in 3D are the class of smooth degree d surfaces. The projection of a smooth degree d surface forms a degree d apparent contour in the image, the apparent contour in this sense is the intersection of the dual of the surface with the image plane [5, 1].

In this paper we consider the class of all degree d surfaces and their associated apparent contours to be representable as degree d hypersurfaces, thus creating a generic form of algebra for their manipulation. This paper will only present a brief theoretical overview of the concepts, although the computationally tractable cases of the apparent contour triangulation and multi-view geometry have been simulated in noise free conditions (up to degree 10). All the geometry and algebra presented in this paper is projective [8]

2 Linear Mutli-View Geometry

This section will outline the notation and the basic building block of linear mulit-view geometry. The development of the ideas underlining multi-view matching constraints and triangulation of linear features is heavily influenced by the notation and stucture of the linear matching constraints presented in [10]. Due to space considerations this paper assumes that the reader is familiar with majority of these concepts.

2.1 Features

The first consideration when dealing with the multi-view geometry of linear features is their notation. Consistantly we will refer to features as any type of geometric object observed in a scene, be this points, lines and planes in the linear case or hypersurfaces in the degree d case.

Table 1 summarises the notation and degrees of freedom (DOF) for the group of linear features in the projective plane ($[A, B, C] \in \mathbb{P}^2$).

Hyperplane	$\mathbb{P}^2$	$\mathbb{P}^{2*}$	DOF
Points	x^A	$x^{[A]} = \epsilon_{ABC} x^A = x_{BC}$	2
Lines	$x^{[AB]}$	$x^{[AB]} = \epsilon_{ABC} x^A = x_{AB}$	1

Table 1. Linear features and there duals in $\mathbb{P}^2$

Similarly, Table 2 summarises the notation and the DOF for linear features in projective space ($[a, b, c, d] \in \mathbb{P}^3$).

Hyperplane	$\mathbb{P}^3$	$\mathbb{P}^{3*}$	DOF
Points	x^a	$x^{[a]} = \epsilon_{abcd} x^a = x_{bcd}$	3
Lines	$x^{[ab]}$	$x^{[ab]} = \epsilon_{abcd} x^{ab} = x_{cd}$	2
Planes	$x^{[abc]}$	$x^{[abc]} = \epsilon_{abcd} x^{abc} = x_d$	1

Table 2. Linear features and there duals in $\mathbb{P}^3$

These tables demonstrate the process of dualization for linear feature types via the antisymmetrization operator $[\ldots]$. The antisymmetrization operator should be considered as a determinantal method to generate the algebra for linear features, by performing an alternating tensor contraction over the space to which the operator is applied [2].

2.2 Triangulation

Triangulation is the process of calculating a feature in $\mathbb{P}^3$ from two or more of its projections in $\mathbb{P}^2$. Firslty, we must consider the projection operator (P^α_β) or *camera matrix* that denotes the projection of linear features from the scene to the image plane ($P^\alpha_\beta : \mathbb{P}^3 \rightarrow \mathbb{P}^3$). Table 3 summarises the range of projection operators for linear features.

Hyperplane	$\mathbb{P}^3$	$\mathbb{P}^{3*}$
Point	$x^A \sim P^A_a x^a$	-
Line	$x^{[AB]} \sim P^{[A}_{[a} P^{B]}_{b]} x^{[ab]}$	$x_{[BC]} \sim P^{[c}_{[B} P^{d]}_{C]} x_{[cd]}$
Plane	-	$x_A \sim P^d_A x_d$

Table 3. Projection operators for linear features

Generally it may be stated that $\lambda x^\alpha = P^\alpha_\beta x^\beta$, where λ is an arbitrary scale factor.

Having observed $2 \ldots n$ image features, triangulation proceeds through the *reconstruction equations*,

$$\begin{pmatrix} P^{A_1}_a & x^{A_1} & 0 & \cdots & 0 \\ P^{A_2}_a & 0 & x^{A_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ P^{A_n}_a & 0 & 0 & \cdots & x^{A_n} \end{pmatrix} \begin{pmatrix} x^a \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} = \mathbf{0} \quad (1)$$

where the resulting nullvector of these equations presents a solution for the scene feature and the scale factors λ_n. The stack of camera matrices on the left hand side of (1) is referred to as the *joint image projection matrix* ($P^\gamma_a \Rightarrow P^{A_1 A_2 \ldots A_n}_a$) and can be thought of as a vector of camera matrices that projects a common feature from the scene (x^a) to its *joint image* feature location ($x^\gamma \Rightarrow x^{A_1 A_2 \ldots A_n}$).

The reconstruction equations have,

$$(\sum_n DOF^n_i - DOF_s - 1)(DOF_s + 1) - n + 1 \quad (2)$$

DOF, where DOF^n_i and DOF_s denote the DOF of the nth image feature and scene features respectively. Furthermore the reconstruction equations are rank-$(DOF_w + n)$.

2.3 Multi-View Constraints

Mutli-View constraints provide a linear relationship between projections of scene features observed in two or more images. Multi-View constraints provide a means to calculate the structure of the scene and the egomotion of the camera. The approach to building multi-view constraints stems from the representation of a subspace in the *Grassmann* algebra. Here we wish to find a d_w-dimensional subspace for the scene (where the scene is embedded in $\mathbb{P}^{d_w}$), from the joint image projection matrix. This is achieved by antisymmetrizing over $d_w + 1$ of the joint images scene indeterminants, with corresponding unique choices of *any* $d_w + 1$ of the images indeterminants,

$$I^{\gamma_0 \cdots \gamma_d} \equiv \frac{1}{(d_w + 1)!} P^{\gamma_0}_a \cdots P^{\gamma_d}_d \epsilon^{a \ldots d} \equiv P^{\gamma_0}_{[a} \cdots P^{\gamma_d}_{d]} \quad (3)$$

(3) is known as the *Joint Image Grassmannian*. The selection of the image indeterminants $\gamma_0 \ldots \gamma_d$ from the rows of the joint image projection matrix determines which images the resulting matching constraint will represent. The choice of rows obeys the simple rules that for an image to be included in the matching constraint, it must be represented by at least one row, and less than $d_i + 1$ rows (where the image plane is embedded in $\mathbb{P}^{d_i}$). This leads to well known set of matching tensors (Table 4) and also explains why there is at most 4-view matching constraints for points and lines.

There are many variations of the atypical matching constraints given in Table 4, see [10, 3].

Views	Tensor	Constraint
2	$I^{B_1C_1B_2C_2}$	$I^{[B_1C_1B_2C_2}x^{A_1}x^{A_2]}=0$
3	$I^{B_1C_1B_2B_3}$	$I^{[B_1C_1B_2B_3}x^{A_1}x^{A_2}x^{A_3]}=\mathbf{0}_{\dots}$
4	$I^{B_1B_2B_3B_4}$	$I^{[B_1B_2B_3B_4}x^{A_1}x^{A_2}x^{A_3}x^{A_4]}=\mathbf{0}_{\dots}$

Table 4. Atypical Linear Matching Constraint Tensors

3 Hypersurfaces

It is at this point that we enter into profitable new territory with the introduction of hypersurfaces into the tensor notation.

Definition A degree d hypersurface is denoted as the d-fold symmetric product (symmetrization) $(\dots)$ of an indeterminant [2]. The resulting hypersurface is considered to be embedded in the space in which the symmetrization operator is applied. That is,

$$\underbrace{x_{(\gamma\dots\gamma)}}_{\mathrm{d-fold}} \tag{4}$$

or in the Algebro-Geometric notation [9],

$$\underbrace{\mathbb{P}^n\times\dots\times\mathbb{P}^n}_{\mathrm{d-fold}}\backslash\mathcal{S}_n^d$$

where $\mathcal{S}_n^d$ is the d-fold symmetric permutation group, this may also be considered as the degree d, n space *Veronese embedding* (ν_n^d).

We can state that hypersurfaces are generically points in a $\mathbb{P}^{\nu_n^d}$ dimensional space, where $\nu_n^d=\binom{n+d}{n}-1$, thus they have ν_n^d-1 DOF. Some common examples of hypersurfaces are the conic $x_{(AA)}x^Ax^A=0$, and the quadric $x_{(aa)}x^ax^a=0$ hypersurfaces. Equivalent dual hypersurfaces are simply $x^{(AA)}x_Ax_A=0$ where $x^{(AA)}\in\mathbb{P}^{\nu_2^2*}$.

4 Degree d Multi-View Geometry of Hypersurfaces

Now we are ready to observe the degree d triangulation and Multi-View Geometry, of hypersurfaces. The development in this section will follow the exact path we took in Section 2, where in this case points and lines will be replaced by hypersurfaces and dual hypersurfaces.

4.1 Triangulation of Hypersurfaces

As in Section 2.1 our first step in solving the triangulation problem is addressing the nature of projection operators for degree d hypersurfaces. However, in this case we are concerned with the degree d embedding of hypersurfaces in $\mathbb{P}^2$ and $\mathbb{P}^3$ respectively. Firslty, we should note that this concept is not completely new, in [4, 1] an equivalent observation was made for the projection of degree 2 hypersurfaces. Table 5 summarises the range of projection operators for degree d hypersurfaces.

	$\mathbb{P}^{\nu_3^d}$	$\mathbb{P}^{\nu_3^d*}$
Hypersurface	$P^{(a\cdots}_{(A\cdots}P^{a)}_{A)}$	$P^{(A\cdots}_{(a\cdots}P^{A)}_{a)}$

Table 5. Projection operators for degree d hypersurfaces

Generally, it may be stated that projection of hypersurfaces is denoted as $\lambda x_{(A\dots A)}=P^{(a\cdots}_{(A\cdots}P^{a)}_{A)}x_{(a\dots a)}$ and dually $\lambda x^{(A\dots A)}=P^{(A\cdots}_{(a\cdots}P^{A)}_{a)}x^{(a\dots a)}$. We can also state that these projection matrices are the d-fold symmetric powers of the regular point projection matrix (and its dual), thus resulting in the dimension of these matrices being $((\nu_3^d+1)\times(\nu_2^d+1))$ and $((\nu_2^d+1)\times(\nu_3^d+1))$ respectively.

Since we are concerned with finding the the equation of the surface generating the apparent contour in the image. We must take the intersection of the dual hypersurfaces tangent cone with the image plane. For notational compactness we will assign $(A_n\cdots A_n)=\eta_n$ and $(a\cdots a)=\mu$. This leads us to the equivalent set of *dual* reconstruction equations for degree d hypersurfaces,

$$\begin{pmatrix} P^{\eta_1}_{\mu} & x^{\eta_1} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ P^{\eta_n}_{\mu} & 0 & \cdots & x^{\eta_n}\end{pmatrix}\begin{pmatrix} x^{\mu}\\ \lambda_1\\ \vdots\\ \lambda_n\end{pmatrix}=\mathbf{0} \tag{6}$$

again the resulting nullvector of these equations presents a solution for the scene hypersurface (x^{μ}) and the scale factors λ_n.

The minimum number of image hypersurfaces required to reconstruct a degree d hypersurface is given as the lower bound of,

$$\nu_3^d\geq n\geq\frac{(d^2+6d+11)}{3(d+3)} \tag{7}$$

[5] the lower bound n must be rounded up to the closest integer value. The upper bound is the limit on the number of images for the resulting matching constraint. The DOF of these reconstruction equations and their rank are analogous to those stated for (1).

4.2 Multi-View Constraints for Degree 2 Hypersurfaces

Before we address the general formulation for the degree d matching constraints for hypersurfaces, we will tred gently by outlining the concepts for degree 2.

Firstly, it is not clear in general the make-up or practical relevance of features embedded in $\mathbb{P}^2$ or $\mathbb{P}^3$ that have DOF other than $\nu_n^d - 1$ (ie. hypersurfaces).

An application of (7) suggests the presence of degree 2 matching constraints for two through to ten image projections. Again, the object in building the matching constraints is select ν_3^2 unique rows from the joint image projection matrix to make up the matching constraints. The corresponding matching constraints are given in Table 6.

Views	Tensor	Constraint
2	$I^{B_1 \cdots F_1 B_2 C_2 D_2 E_2 F_2}$	$I^{[\cdots} x^{A_1} x^{A_2]} = 0$
3	$I^{B_1 \cdots F_1 B_2 C_2 D_2 E_2 B_3}$	$I^{[\cdots} x^{A_1} x^{A_2} x^{A_3]} = \mathbf{0} \ldots$
4	$I^{B_1 \cdots F_1 B_2 C_2 D_2 B_3 B_4}$	$I^{[\cdots} x^{A_1} x^{A_2} x^{A_3} x^{A_4]} = \mathbf{0} \ldots$
$\vdots$	$\vdots$	$\vdots$
10	$I^{B_1 B_2 \cdots B_{10}}$	$\ldots$

Table 6. Degree 2 Matching Constraint Tensors

It is not clear what the actual effect of selection of different rows from the joint image has on the resulting matching constraint (future work may included a thorough investigation of this uncertainty along the lines of [**?**]). But from the initial experimentation we have found that any combination of rows that meets the aforementioned requirements for defining a Grassmann subspace is adequate to construct the matching tensor. The most pertinent factor in selecting a number of rows to form the matching constraints, is minimising the size of the actual matching tensor.

The selection of k rows from a space of size n will result in the size of associated dimension of the matching tensor being $\binom{n}{k}$, so naturally values close to either n or 1 will yield smaller matching constraints.

4.3 Multi-View Constraints for Degree d Hypersurfaces

Finally, we can now see that an application of equation (7) will give the upper and lower bounds for the degree d multi-view constraints and an application of equation (6) will generate the reconstruction equations for the problem. Any selection of rows from the reconstruction equations meeting the aforementioned criteria of a valid subspace, will be adequate to reconstruct the degree d matching constraints.

5 Conclusions and Future Work

The authors have presented a general closed form linear method for the solution of degree 2 curves and surfaces which extends to the solution of degree d surfaces. The essential problems of triangulation and multi-view matching constraints for these features have been considered, unfortunately due space restrictions a full account of these geometries has been limited.

The authors have also considered a practical scheme to calculate the apparent contours through the fitting of cubic NURBS curves. NURBS are the only alternative since they have the essential property of projective closure. Once the NURBS curves have been fitted to the image data they must then be converted into their *implicit* representation via the process of implicitization [7]. This topic will be considered in future work.

References

[1] G. Cross. *Surface Reconstruction from Image Sequences : Texture and Apparent Contour Constraints*. PhD thesis, University of Oxford, Trinity College, 2000.

[2] W. Greub. *Multilinear Algebra*. Springer-Verlag, 1978.

[3] R. I. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, 2000.

[4] F. Kahl and A. Heyden. Using conic correspondences in two images to estimate the epipolar geometry. *Int. Conf. on Computer Vision*, 1998.

[5] J. Y. Kaminski, M. Fryers, A. Shashua, and M. Teicher. Multiple view geometry of non-planar algebraic curves. *Int. Conf. on Computer Vision*, 2001.

[6] L. Quan. Conic reconstruction and correspondence from two views. *Transactions on Pattern Analysis and Machine Intelligence*, 18(2), 1996.

[7] T. W. Sederberg. Applications to computer aided design. In *Proc. of Symposia in Applied Mathematics*, volume 53, pages 67–89. AMS, 1998.

[8] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[9] I. R. Shafarevich. *Basic Algebraic Geometry*. Springer-Verlag, 1974.

[10] W. Triggs. The geometry of projective reconstruction i: Matching constraints and the joint image. *Int. Conf. on Computer Vision*, pages 338–343, 1995.